An Evaluation of Computational Algorithms in the Automatic Generation of Zonal Reduced-Order Models from CFD Simulations

Thibault Marzullo[1], Marcus M. Keane[2], Marco Geron[3], Rory F.D. Monaghan[1]
[1]Department of Mechanical Engineering, National University of Ireland, Galway
[2] Department of Civil Engineering, National University of Ireland, Galway
[3]School of Mechanical and Aerospace Engineering, Queen's University Belfast

Abstract

The present study compares the accuracy and time to solution of three different algorithms used in automatic Reduced-Order Model (ROM) generation. This is part of an effort to develop a systematic tool chain for automatically extracting accurate ROMs from Computational Fluid Dynamics (CFD) simulations for use in the design and operation of near-zero energy buildings, with a higher accuracy than traditional zonal models but at a fraction of the computational cost of CFD. The results show that the Mean Values Segmentation (MVS) method is preferable to Coarse Grid Interpolation (CGI) and Classic Watershed (CW) for this application. Furthermore, it is possible to generate accurate models with mean absolute errors for temperature distributions as low as 0.3 K that can be generated in less than 70 seconds and solved in less than 1 second. Further work is needed to develop the flexibility of the tool chain to solve for Boundary Conditions (BC) different from the initial CFD BC's. This will allow the generation of multiple ROMs from a reduced set of CFD simulations, allowing to accurately simulate indoor thermal conditions at low computational costs.

1 Introduction

Sustainability, as defined by the ASHRAE Handbook of Fundamentals (ASHRAE, 2013), is "providing for the needs of the present without detracting from the ability to fulfil the needs of the future". A key approach to improve the sustainability of energy use is the reduction of waste through careful planning, optimization and management of demand. In the built environment, the use of models of building thermal conditions during both the design and operation phases is recognised as a powerful method to increase energy efficiency and reduce energy demand. These computer models can be grouped into three categories: (1) physics-based models, (2) experimental models, and (3) mixed models (Harish and Kumar 2016). Physics-based (theoretical) models describe in detail the studied system and its subsystems, and define an output based on mathematical equations constrained by physical laws. They are typically used during the preliminary design and energy audit phases. Experimental models on the other hand are tailored to a particular system. Through experimentation, the system's response to various inputs is evaluated and the model is developed accordingly. Finally, mixed models are physics-based models for which parameters are estimated using statistical and/or experimental analysis. Among physics-based models, Computational Fluid Dynamics (CFD) is a powerful and increasingly widespread tool for simulating fluid domains, which yields models of high fidelity (van Hooff and Blocken 2013; Tamura and Van 2014). However, its large computational expense renders it too time-consuming for operational, and even some design, tasks. Designers and operators therefore often choose lower fidelity but more efficient methods when available (Li and Wen 2014), such as RC models.

In order to overcome some of these issues we propose a tool chain called CFD-ROM (Computational Fluid Dynamics – Reduced Order Model), which will automatically extract ROMs from CFD simulations and solve them rapidly for a wide range of conditions, including those for which no CFD solutions are available (Marzullo et al. 2017). The method aims to develop ROMs that will retain a high level of accuracy, similar to the CFD simulations, but at a significantly lower computational cost (Mora, Gadgil, and Wurtz 2003; Lucia, Beran, and Silva 2001). The purpose of this study presented in this paper is to perform an evaluation of different ROM generation algorithms to inform the development of the CFD-ROM tool chain.

2 Methods

The CFD-ROM method is described in detail in Marzullo et al. (2017), and for the sake of brevity is summarised here. The method uses results obtained from a validated CFD simulation as input (step 1). It then clusters computational cells together to create zones (step 2). Zones' physical properties, boundary conditions and inter-zone mass-flow and heat transfer interactions are calculated (step 3). Afterwards, a multi-zone ROM is generated and passed to the ROM solver (step 4). It is solved (step 5) and the results are remapped to the original CFD domain (step 6).

A crucial step in the development of the CFD-ROM method is the selection of a zone generation algorithm to develop a multi-zone ROM. Three algorithms are considered in the present study: (1) Mean Values Segmentation (MVS), (2) Classic Watershed (CW), and (3) Coarse Grid Interpolation (CGI). All three algorithms, which are explained in the following sections, have been coded in Python v2.7 (Python Software Foundation., n.d.).

This section briefly describes the methods of each algorithm. Further details are available in the previous study by the authors (Marzullo et al. 2017), which made an initial comparison of the algorithms effectiveness to generate input files for ROM solvers. The present study brings the comparison to the next stage by comparing the accuracy and time to solution of the ROMs generated using the three algorithms.

2.1 Zone generation algorithms

MVS successively splits the domain along the mean values of selected zone criteria. Zone criteria are the physical properties that are used to identify uniform regions of the simulation domain. Examples of zone criteria include temperature, air velocity and carbon dioxide concentration. For the purpose of the present comparative study, temperature is the only zone criterion used. Zone generation is achieved through three main steps described by Marzullo et al. (2017): (1) zone-type generation, (2) zone creation, and (3) zone number reduction. A zone-type is an interval containing all the cells sharing similar values of the zone criteria regardless of their spatial position. Zones are defined as spatially discrete regions of common zone-type.

CW is a method used in image processing for image segmentation (Soille and Vincent 1990). The authors adapted it to ROM generation because its principle is very similar to zone creation. CW comprises 4 steps described in detail by Soille and Vincent (1990): (1) a height map is generated from the gradient of temperature in the domain; (2) working only with the height map, a water level is defined. The water level will rise (3) at each new iteration, and the data points belonging to the same basin (i.e. volumes with small temperature gradients) are clustered together (4). Basins will grow until either they encounter another basin or a maximal height is reached. Once the height map has been iteratively flooded, a set of clusters of cells (zones) belonging to the same basin is obtained.

In CGI, the CFD-predicted property fields are interpolated from the CFD mesh to a much coarser mesh. For this study, the mesh was coarsened from 1.5 million cells to 8–63 zones (see the Results section) based on the X, Y, Z coordinates of each cell. The algorithm developed for this study (1) scans the domain and defines a coarse mesh based on the desired number of zones, and (2) assigns cells to a zone depending solely on their coordinates.

2.2 ROM solver

The ROMs generated by each algorithm are passed to an external solver, Sinda/FLUINT (Baumann and Cullimore 2002), which is a one-dimensional solver for thermal and hydraulic lumped models. The ROM is expressed in the form of a fluid network in which (1) zones are considered uniform lumps with a constant volume, (2) inlets and outlets are represented by uniform lumps of infinite capacity, (3) thermal boundaries such as walls are represented by thermal lumps with constant temperature and thermal transfer coefficient, and (4) the interactions between zones are represented by fixed mass flow rates between fluid lumps. The model can then be solved, and the result is remapped back to the original CFD domain using the Python script. The error between the ROM and the CFD results is then computed using the method described below.

Results

Comparisons of the time to generation and accuracy of ROMs generated with the three algorithms are presented in this section. The basis of the accuracy comparison is the fidelity of the solved ROMs against original validated CFD data. The present study uses a weighted Mean Absolute Error or WMAE in units of K, as shown in Equation (1).

$$\boldsymbol{WMAE} = \sum_{i=o}^{n} \frac{V_i \times |T_{CFDi} - T_i|}{n \times V_{domain}} \quad (1)$$

Where *n* is the number of cells in the domain, V_i is the volume of cell i, V_{domain} is the total volume of the domain, T_{CFDi} is the original CFD temperature of cell i, and T_i is the temperature assigned to the cell after solving the ROM.

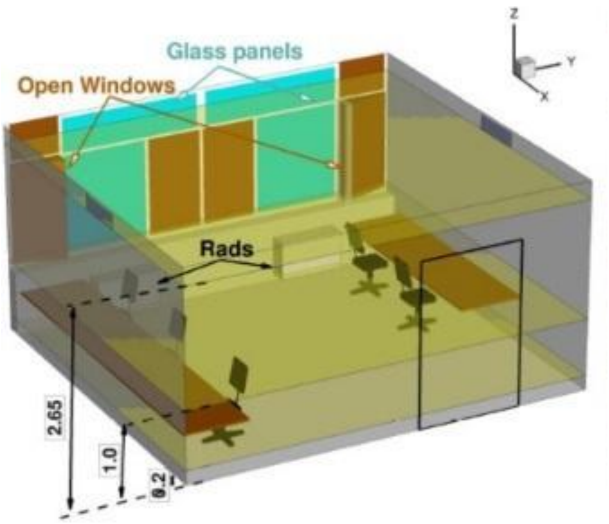

Figure 1 Numerical domain of the LG04 office in the ERI building of UCC, which CFD simulation is used in this study.

The data used in this comparative study are taken from the previously-validated CFD model of a north-facing office in the Environmental Research Institute (ERI) building at University College Cork (UCC) (Mullen et al., 2015) as shown in **Figure 1**. CFD models were developed using Phoenics (Rosten, Spalding, and Tatchell 1983) modelling software to generate a database of test cases. For all simulations, turbulence is modelled using the steady-state Reynolds Average Navier-Stokes (RANS) approach coupled with the Re-Normalisation Group

(RNG) k-ε turbulence model. Air is modelled as an incompressible ideal gas. Phoenics utilises an immersed body technique and consequently the domain is discretized using a Cartesian structured grid with 1,572,165 cells (115x147x93). Constant temperature boundary conditions have been utilised for the ceiling, the floor, and east and west walls. All other objects are considered adiabatic. All CFD simulations used have been validated with experimental data and previously published (Mullen et al. 2015). All the multi-zone models used to obtain the results are automatically extracted using the three algorithms (MVS, CW and CGI) with the only input from the user being the required number of zones.

The following results have been obtained for ROMs consisting of N = 4-43 zones for MVS, N = 8-60 zones for CW, and N = 8-63 zones for CGI. *Figure 2* shows WMAE versus number of zones N for the three algorithms. *Figure 3* shows time to generation t_{gen} (in seconds) versus number of zones N for the three algorithms.

The results indicate that for this range of numbers of zones, CGI produces multi-zone outputs in which accuracy is dependent on the geometry of the coarse grid. In fact, the more accurate results (for 9, 13, and 26 zones, as shown in *Figure 2*) correspond to grids that were divided more finely in the vertical direction in order to better capture thermal stratification. The mean error of the outputs produced with CGI usually converges for larger numbers of zones than the other algorithms. The results obtained for CW indicate that the zone-generation method of the algorithm is not optimal. The algorithm needs an initial set of "seeds" to start the zoning process, which are extracted by detecting peaks (maxima and minima) in temperature.

The algorithm generates and expands zones starting with these seeds. Future work includes the investigation of techniques that could ensure the optimal seeds are selected for better accuracy.

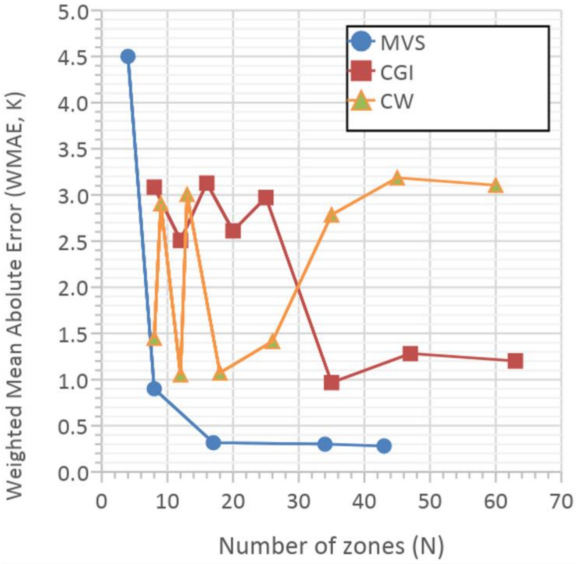

Figure 2: WMAE versus number of zones of solved ROMs generated with the MVS, CGI and CW zone generation algorithms

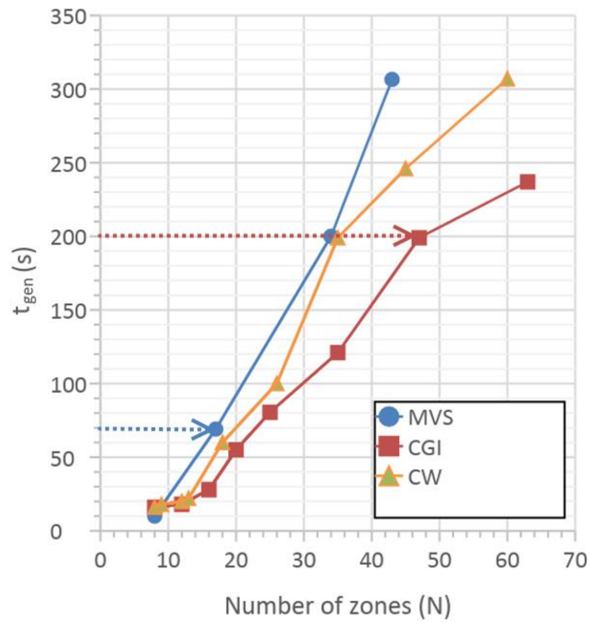

Figure 3: Time to generation of a complete ROM vs number of zones of solved ROMs generated with the MVS, CGI and CW algorithms. The number of zones at which the solution reaches convergence is indicated for MVS and CGI.

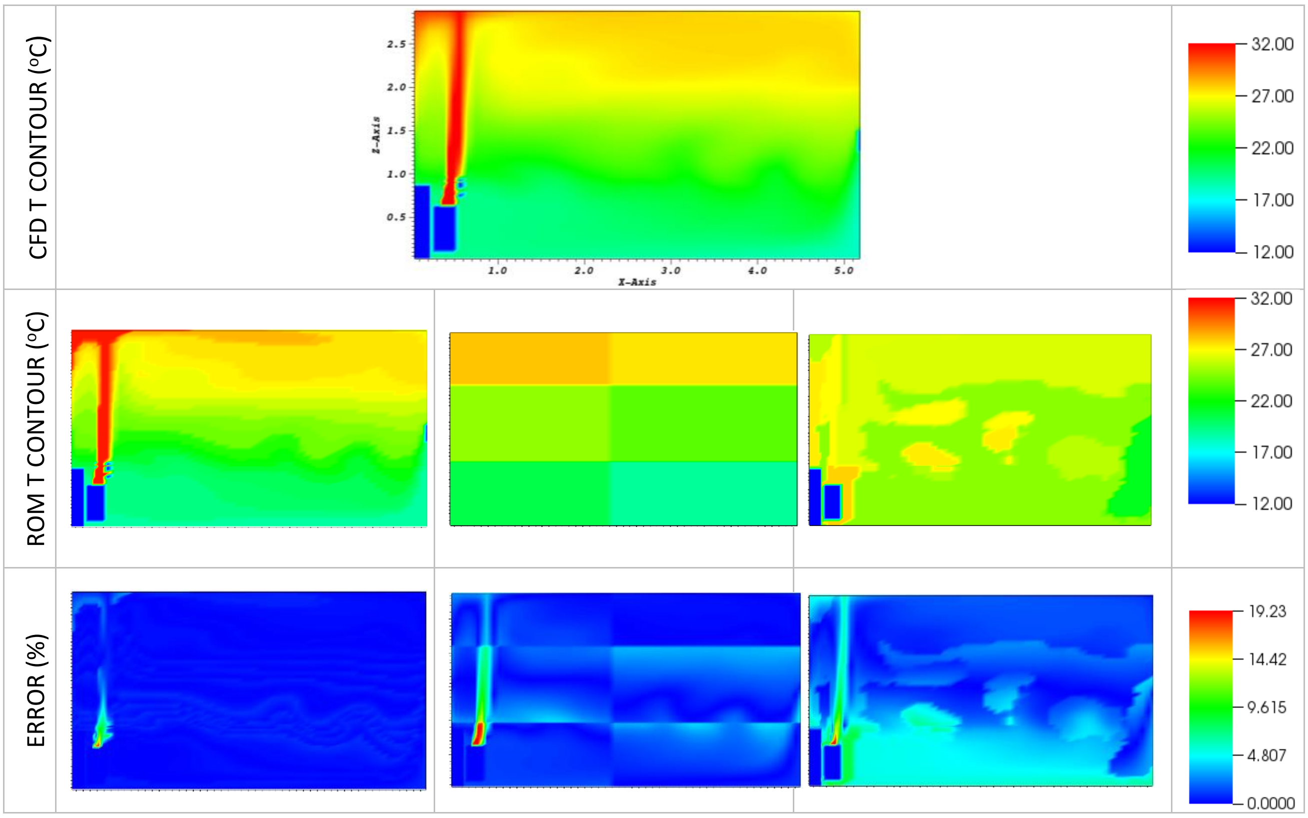

Figure 4: Temperature plots of solved ROMs versus original CFD temperature, and error plots for MVS (left), CGI (centre) and CW (right)

MVS produces ROMs which solution converges at a smaller number of zones and to a lower error than the other algorithms.

The time to solution is the same for all the algorithms and varies from 0.9s for 8-zone ROMs up to 1.2s for 34-zone ROMs, against 2hours and 11 minutes for the original CFD simulation. CGI generates ROMs faster than the other methods for the same number of zones, but their solution converges at larger numbers of zones thus increasing the time to generate a viable ROM. ROMs produced with MVS reach convergence at 16 zones which are generated in under 70s. CW does not clearly reach convergence at 60 zones, which is far more than the previous methods. Figure 4 shows temperature contours for the original CFD results, and 17-zone ROMs generated using the three algorithms as well as error contour plots. It shows MVS's ability to capture zones with high thermal gradients as well as air stratification, while CW and CGI fail to capture high gradients, and shows CGI's high dependence on the re-meshing parameters.

Discussion

The results indicate that it is possible to reach ROM zone number independence around 17 zones for MVS, along with a WMAE of 0.3K. The algorithm is able to capture temperature gradients around convectors and air stratification, which suggests that it can be used to generate ROMs for modelling the built environment. Future work will include the assessment of the algorithm performance in case studies featuring air stratification, naturally ventilated indoor spaces, etc. which can be met in the built environment. On the other hand, the error of ROMs generated with CGI's converges at higher numbers of zones, requiring more time to generate the ROMs which still have lower accuracy than MVS. Finally, CW has proven unsuccessful, as the accuracy is highly dependent on the case study and generally produces results less accurate than the other algorithms.

Conclusions

The main objective of this study was to compare zone generation algorithms to be used in a novel CFD-ROM method for thermal modelling of indoor environments. At this stage, MVS was found to be the most suitable method because of its ability to efficiently capture thermal distributions at a lower computational cost than the other algorithms, while preserving most of CFD simulation's accuracy. In fact, the weighted mean average error quickly dropped below 0.3 K for a 17-zone ROM that can be generated and solved in under 70s. Next steps include (1) testing the CFD-ROM toolchain in a range of different CFD test cases to assess its flexibility, (2) solving the ROM for BCs that differ from the original CFD BCs in order to generate more ROMs than there are CFD simulations, and (3) developing a Modelica solver.

Acknowledgements

The authors wish to acknowledge the following funding. T. Marzullo is supported by a Scholarship from the College of Engineering and Informatics at NUI Galway and by HIT2GAP (EU/H2020 Grant Agreement No.: 680708).

References

Baumann, Jane, and Brent Cullimore. 2002. "Nonlinear Programming Applied to Calibrating Thermal and Fluid Models to Test Data." In Semi-Therm. www.crtech.com.

Harish, V.S.K.V., and Arun Kumar. 2016. "A Review on Modeling and Simulation of Building Energy Systems." Renewable and Sustainable Energy Reviews 56: 1272–92. doi:10.1016/j.rser.2015.12.040.

Li, Xiwang, and Jin Wen. 2014. "Review of Building Energy Modeling for Control and Operation." Renewable and Sustainable Energy Reviews 37. Elsevier: 517–37. doi:10.1016/j.rser.2014.05.056.

Lucia, David J., Philip S. Beran, and Walter A. Silva. 2001. "Reduced-Order Modeling - New Approaches for Computational Physics." 39th Aerospace Sciences Meeting and Exhibit 40 (1–2). American Institute of Aeronautics and Astronautics: 51–117. doi:10.2514/6.2001-853.

Marzullo, T., Yousefian, S., Keane, M.M., Geron, M., and Monaghan, R.F.D. 2017. "A Comparative Study of Computational Algorithms Used in the Automatic Generation of Reduced-Order Models from CFD Simulations." In 3rd Building Simulation Applications Conference Proceedings. Unibz.

Mora, L, A J Gadgil, and E Wurtz. 2003. "Comparing Zonal and CFD Model Predictions of Isothermal Indoor Airflows to Experimental Data." Indoor Air 13 (2). Wiley-Blackwell: 77–85. doi:10.1034/j.1600-0668.2003.00160.x.

Mullen, D.T., M.M. Keane, M. Geron, and R.F.D. Monaghan. 2015. "Automatic Extraction of Reduced-Order Models from CFD Simulations for Building Energy Modelling." Energy and Buildings 99 (July): 313–26. doi:10.1016/j.enbuild.2015.04.015.

Python Software Foundation. n.d. "Python Language Reference, Version 2.7."

Rosten, HI, DB Spalding, and DG Tatchell. 1983. "PHOENICS: A General-Purpose Program for Fluid-Flow, Heat Transfer and Chemical-Reaction Processes." CHAM.

Soille, Pierre, and Luc Vincent. 1990. "Determining Watersheds in Digital Pictures via Flooding Simulations." Visual Communications and Image Processing 1360: 240–50.

Tamura, Yukio, and Phuc Pham Van. 2014. "Development of CFD and Applications : Monologue by a Non-CFD-Expert." CWE_Global Overview of Wind Engineering 144: 3–13. doi:10.1016/j.jweia.2015.05.003.

van Hooff, T., and B. Blocken. 2013. "CFD Evaluation of Natural Ventilation of Indoor Environments by the Concentration Decay Method: CO2 Gas Dispersion from a Semi-Enclosed Stadium." Building and Environment 61 (March): 1–17. doi:10.1016/j.buildenv.2012.11.021.